



Wireless Sensor Networks for Habitat Monitoring

Alan Mainwaring, Joseph Polastre, Robert Szewczyk, and David Culler

IRB-TR-02-006

June, 2002

DISCLAIMER: THIS DOCUMENT IS PROVIDED TO YOU "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE. INTEL AND THE AUTHORS OF THIS DOCUMENT DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OR IMPLEMENTATION OF INFORMATION IN THIS DOCUMENT. THE PROVISION OF THIS DOCUMENT TO YOU DOES NOT PROVIDE YOU WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS

Wireless Sensor Networks for Habitat Monitoring

Alan Mainwaring¹

Joseph Polastre²

Robert Szewczyk²

David Culler^{1,2}

¹ Intel Research Laboratory, Berkeley
Intel Corporation
{amm,dculler}@intel-research.net

² EECS Department
University of California at Berkeley
{polastre,szewczyk,culler}@cs.berkeley.edu

Abstract— We examine the requirements of environmental monitoring in the context of two wildlife habitats: Great Duck Island and James Reserve. Based on the requirements from the researchers studying these habitats, we propose a sensor network architecture for this class of applications, discuss the hardware design: sensor platform, enclosure design, and sensor calibration. Available energy emerges as the resource dictating performance characteristics of various services: data collection, communication, sensor network retasking. We evaluate the tradeoffs between different approaches to implementing several sensor network services. The ultimate goal of our work is to provide life scientists with a reliable and predictable sensor network kit.

I. INTRODUCTION

Habitat and environmental monitoring represent a class of sensor network applications with enormous potential benefits for scientific communities and society as a whole. Our technical interests in these applications are four fold. First, they focus attention on developing an appropriate sensor network architecture for a domain rather than an idiosyncratic instance. Second, they provide a context in which some problems have simple, concrete solutions while others remain open research areas. Third, an application-driven approach separates actual problems from potential ones, and relevant issues from irrelevant ones. Finally, collaboration with scientists in other fields helps to define the broader application space as well as specific application requirements, allows field testing emerging systems, and offers objective evaluations of the technologies. The impact of sensor networks for habitat and environmental monitoring will be measured by their ability to enable of new applications and production of new results that would otherwise be too difficult to realize.

The instrumentation of natural spaces with networked sensors enables long-term data collection at scales or resolutions that are difficult, if not impossible, to obtain otherwise. The intimate connection with their immediate physical environments allows sensor networks

to provide localized measurements and detailed information that complement the macroscopic measurements and analysis. The integration of on-board processing, local storage and networking allows individual sensor nodes to perform complex filtering and triggering functions, and apply application- and sensor-specific data compression algorithms. When local processing is combined with cooperative in-network processing, more complex tasks, like statistical sampling, data aggregation, and system health and status monitoring, become possible [1], [2].

Several qualitative differences from traditional instrumentation make sensor networks attractive for habitat and environmental monitoring. The complete integration of computing, networking and local storage with sensing and actuation produces smaller, low-power devices. Increased power efficiency gives applications more flexibility in resolving fundamental design tradeoffs, *e.g.* between sampling rates and battery lifetimes. Low-power radios with well-designed protocol stacks allow generalized communications among network nodes, rather than simple point-to-point telemetry. The computing and networking capabilities allow sensor networks to be reprogrammed or retasked after deployment in the field. Moreover, nodes have the capability to adapt their operation over time in response to changes in the environmental as well as the condition of the sensor network itself.

This paper develops a specific habitat monitoring application, that is representative of habitat and environmental monitoring applications in general. It presents a collection of requirements, constraints and guidelines that serves as a basis for the resulting sensor network architecture in the real-world. It describes the core components of the sensor network for this domain – the hardware and sensor platforms, patch gateways, basestations and databases. The design and implementation of the essential network services, including power management, communications, retasking and node management, can be evaluated in context.

The remainder of the paper is organized as follows.

Section II discusses the requirements of our habitat monitoring application. Section III presents a sensor network architecture that includes the core system components and interfaces. Section IV discusses the design and implementation issues facing the hardware and software modules. Section V presents the required network services in the light of available energy. Section VI documents our progress and experiences to date in developing and deploying two sensor networks in the field, and section VII makes concluding remarks.

II. APPLICATION REQUIREMENTS

The challenge from a systems research perspective is finding representative applications that further the development of a sensor network architecture and focuses attention on the core system issues. Equally important, research prototypes must be sufficiently robust to allow colleagues in the life sciences to use them to good effect. Multi-disciplinary collaborations are essential for identifying core application requirements, as well as for assisting with the deployment of prototype systems, long-term usage and monitoring, and providing objective feedback on their strengths and weaknesses.

A. Field Stations and Research Overviews

We have selected two locations for field testing in-situ sensor networks for habitat monitoring. Both sites are associated with institutions with ongoing field research programs that have well established on-site infrastructure and logistical support.

The first location is Great Duck Island (GDI). Great Duck Island (44.09N, 68.15W) is a 237 acre island located 15 km south of Mt Desert Island, Maine. The Nature Conservancy and the State of Maine hold much of the island in joint tenancy. Research on their property is conducted under a cooperative agreement with the College of the Atlantic in Bar Harbor, Maine.

Ongoing research on Great Duck Island focuses on basic ecology – the distribution and abundance of plants and animals – in relation to the diverse assortment of micro-climates and habitats. Of particular interest are large breeding colonies of Leech’s Storm Petrels and other seabirds. Sensor networks that measure basic environmental parameters such as light, temperature, humidity, and pressure will provide long-term, baseline data. Infrared sensors may capture individual entrance/exit events as birds move in and out of their nesting burrows in the ground. In the future, additional sensors will support specific studies, such as using a small washdown scale placed inside a sample of burrows for monitoring nest occupancy and egg development.

The second location is the James San Jacinto Mountains Reserve (JMR), Idyllwild, California. The James Reserve (33.48N, 116.46W) is a 29 acre ecological preserve, representing just one of the University of California System Natural Reserve System’s 34 land holdings. These reserves are available for university-level courses, research, and public outreach programs.

As part of the NSF Center for Embedded Networked Sensors, research at the James Reserve is investigating sensing infrastructures for a range of habitats, from streambeds to forests to desert landscapes. They are exploring multimedia sensors for both natural and artificial enclosures, such as nest boxes and bat caves. Other work will focus on monitoring ecosystems over time, including the response of vegetation to climate changes. Additional work will explore acoustical sensing of birds for identification as well as estimating populations.

B. General application requirements

1) *Internet access*: The sensor networks at GDI and JMR must be accessible via the Internet. An essential aspect of habitat monitoring applications is the ability to support remote interactions with in-situ networks.

2) *Hierarchical network*: The field stations at GDI and JMR need sufficient resources to host Internet connectivity and database systems. However, the habitats of scientific interest are located up to several kilometers further away. A second tier of wireless networking provides connectivity to multiple patches of sensor networks deployed at each of the areas of interest. Three to four patches of 100 static (not mobile) nodes is sufficient to start.

3) *Sensor network longevity*: Sensor networks that run for 9 months from non-rechargeable power sources would have significant audiences today. Although ecological studies at GDI and JMR span multiple field seasons, individual field seasons typically vary from 9 to 12 months. Seasonal changes as well as the plants and animals of interest determine their durations.

4) *Operating off-the-grid*: Every level of the network must operate with bounded energy supplies. Although renewable energy, for example solar power, may be available at some locations, disconnected operation remains a possibility. Both GDI and JMR have sufficient solar power to run many elements of the application 24x7 with low probabilities of service interruptions due to power loss.

5) *Management at-a-distance*: The remoteness of the field sites requires the ability to monitor and manage sensor networks over the Internet. Although personnel may be on site (personnel is available year round at JMR but just 2 to 3 months each summer at GDI), the goal is zero

on-site presence for maintenance and administration during the field season, except for installation and removal of nodes.

6) *Inconspicuous operation*: Habitat monitoring infrastructure must be inconspicuous. It should not disrupt the natural processes or behaviors under study. Removing human presence from the study areas both eliminates a source of error and variation in data collection, as well as a significant source of disturbance.

7) *System behavior*: From both a systems and end-user perspective, it is critical that sensor networks exhibit stable, predictable, and repeatable behavior whenever possible. An unpredictable system is difficult to debug and maintain. More importantly, predictability is essential in developing trust in these new technologies for life scientists.

8) *In-situ interactions*: Although the majority of interactions with the sensor networks are expected to be via the Internet, local interactions are required during initial deployment, during maintenance tasks, as well as during on-site visits. PDAs serve an important role in assisting with these tasks. They may directly query a sensor, adjust operational parameters, or simply assist in locating devices.

9) *Sensors and sampling*: For our particular applications, the ability to sense light, temperature, infrared, relative humidity, and barometric pressure provide an essential set of useful measurements. The ability to sense additional phenomena, such as acceleration/vibration, weight, chemical vapors, gas concentrations, pH, and noise levels would augment them.

C. Data models

Although a large number of sensors can easily produce more data than the network can deliver to a relay node or the site's basestation, archiving sensor readings for off-line data mining and analysis is essential. The reliable offloading of sensor logs to databases in the wired, powered infrastructure is an essential capability. The desire to interactively "drill-down" and explore individual sensors, or a subset of sensors, in near real-time complement log-based studies. In this mode of operation, the timely delivery of fresh sensor data is key. Lastly, nodal data summaries and periodic health-and-status monitoring requires timely delivery.

III. SYSTEM ARCHITECTURE

Having defined application requirements we now describe the system architecture, the functionality of individual pieces and how they address the requirements set forth in section II.

The lowest level of the sensing application is provided by autonomous *sensor nodes*. These small, battery-powered devices are placed in areas of interest. Each sensor node consists of two logical components: (1) a general purpose computational module and (2) an application-specific sensing module. This separation makes the platform more flexible because different habitats may require different sensor suites.

Each sensor node collects environmental data primarily about its immediate surroundings. Because it is placed close to the phenomenon of interest, the sensors can often be built using smaller and cheaper individual sensors. High spatial resolution can be achieved through dense deployment of sensor nodes. Compared with an approach which uses a few high quality sensors with sophisticated signal processing, this architecture provides higher robustness against occlusions and component failures.

The general purpose computing module is a programmable unit that provides computation, storage, and bidirectional communication with other nodes in the system. It interfaces with the analog and digital sensors on the sensor module, performs basic signal processing (*e.g.* simple translation based on calibration data), and dispatches the data according to the application needs. Compared with the traditional data logging systems, it offers two advantages: it can be retasked in the field and it can easily communicate with the rest of the system. In-situ re-tasking allows the scientists to refocus their observations based on the analysis of the initial results. For example, while at the beginning of the deployment, we might want to collect the absolute temperature readings, after the initial interpretation of the data we might realize that the information of interest is contained in significant temperature changes that exceed a defined threshold over time.

Individual sensor nodes communicate and coordinate with one another. The cooperation can take several forms. The sensors will typically form a multihop network by forwarding each other's messages, which vastly extends connectivity options. If appropriate, the network can perform in-network aggregation (*e.g.* reporting the average temperature across a region). This flexible communication structure allows us to produce a network which delivers the required data while meeting the energy requirements. We expand on energy efficient communication protocols in section V.

Ultimately, the data from each sensor needs to propagate to the Internet. Bringing direct wide area connectivity to each sensor path is not feasible – the equipment is too costly, it requires a lot of power and the installation all required required equipment is quite intrusive to the habitat. Instead, the wide area connectivity is brought to the *base*

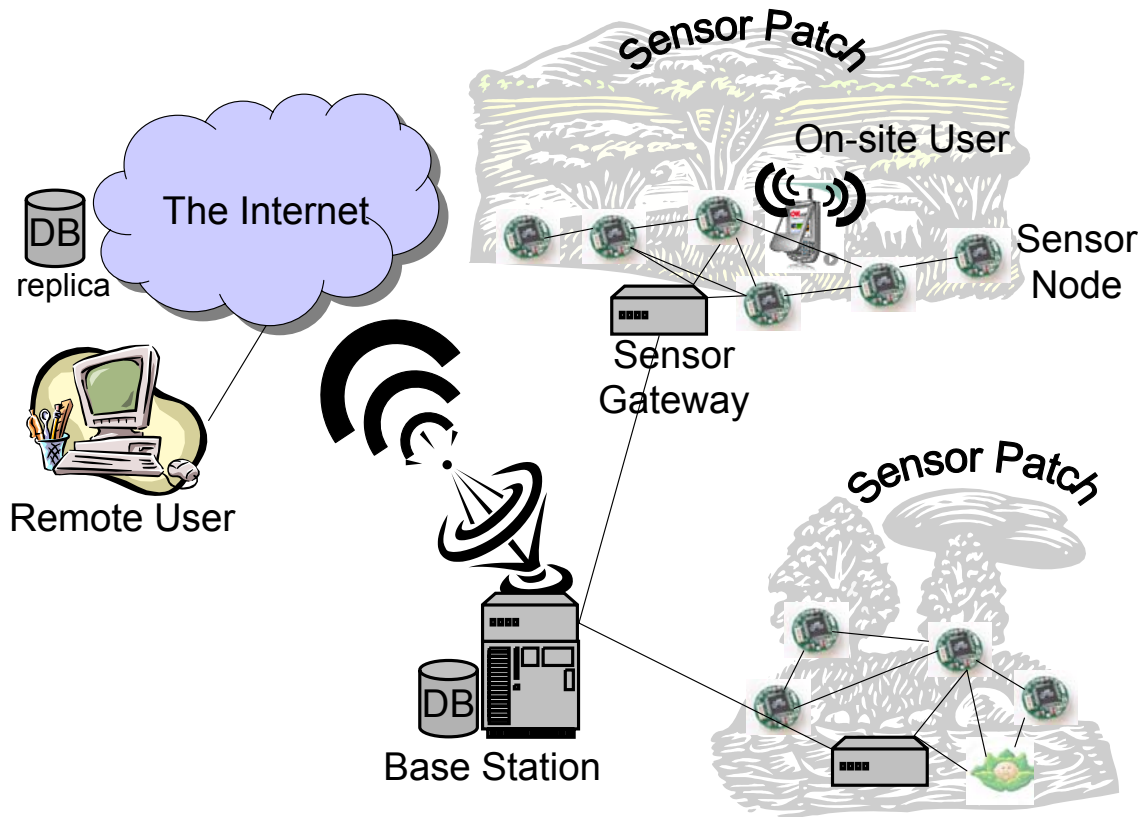


Fig. 1

SYSTEM ARCHITECTURE FOR HABITAT MONITORING

station, where we can provide adequate power and housing for the equipment. The base station communicates with the sensor patch using a wireless local area network. Such design is particularly advantageous since often each habitat involves monitoring several particularly interesting areas, each with its own dedicated sensor patch. Each sensor patch is equipped with a *gateway* which can communicate with the sensor network and provides commercial WLAN connectivity. The WLAN access point is co-located with the base station. In addition to providing the LAN connectivity, the gateway coordinates the activity within the sensor patch, and provides additional computation and storage. The extra resources at the gateway come at a cost – providing enough energy to sustain the unit pushes its size to roughly the size of a car battery.

A valid alternative would be to provide the local area connectivity through the sensor nodes. One would simply place a series of nodes along the path between the sensor patch and the base station; each node in the series acts as a relay. Robustness of such link could be achieved through a redundant connectivity. Each design has different characteristics with respect to expected robustness, bandwidth, energy efficiency, cost, and manageability. We do note that

this alternative only addresses link-level issues; a more powerful gateway node still has an important role to play.

To provide data to remote end-users, the *base station* includes WAN connectivity and persistent data storage for the collection of sensor patches. Since many habitats of interest are quite remote, we expect that the WAN connection will be wireless (e.g. two-way satellite). The base station will typically take a form of a WAN connection, interfaces to the sensor network gateways, a persistent storage component, and a general-purpose computer. The set of components needs to be reliable, enclosed in environmentally protected housing, and provided with adequate power. In many environments such conditions can be provided relatively easily at a ranger station.

The architecture needs to address the possibility of disconnection at every level. Each layer (sensor nodes, gateways, base stations) has some persistent storage which protects against data loss in case of power outage. Each layer also provides data management services. At the sensor level, these will be quite primitive, taking the form of data logging. The base station will often offer full-fledged relational database service. The data management at the gateways will fall somewhere in between, offering

some database services, but perhaps over limited window of data. While many types of communication can be unreliable, when it comes to data collection, long-latency is preferable to data loss. For this kind of communication, a “custody transfer” model, similar to SMTP messages or bundles [3], may be applicable.

Users interact with the sensor network data in two ways. Remote users access the replica of the base station database (in the degenerate case they interact with the database directly). This approach allows for easy integration with data analysis and mining tools, while masking the potential wide area disconnections with the base stations. Remote control of the network is also provided through the database interface. Although this control interface is sufficient for remote users, on-site users may often require a more direct interaction with the network. A small, PDA-sized device, referred to as *gizmo* enables such interaction. The *gizmo* can directly communicate with the sensor patch, provide the user with a fresh set of readings about the environment and monitors the network. While the *gizmo* will typically not take custody of any data, it allows the user to interactively control the network parameters by adjusting the sampling rates, power management parameters and other network parameters. The connectivity between any sensor node and the *gizmo* does not have to rely on functioning multihop sensor network routing, instead the user will often communicate with the mote network directly, relying on single hop proximity. We expect that this device will be extremely useful during the initial deployment and during retasking of the network.

IV. DESIGN AND IMPLEMENTATION STRATEGIES

A. Sensor Network Node

In our deployment, we are using UC Berkeley *motes* as the sensor nodes. We are relying on the latest member of the mote family called Mica [4], which provides several improvements over the previous generations. Mica uses a single channel, 916MHz radio from RF Monolithics to provide a bi-directional communication at 40kbps, an Atmel Atmega 103 microcontroller running at 4MHz, and considerable amount of nonvolatile storage (512 KB). A pair of conventional AA batteries and a DC boost converter provide a stable voltage source, though other renewable energy sources can be easily used. Small size (approximately 2.0 x 1.5 x 0.5 inches) and wireless communication capabilities allow us to deploy the motes in remote locations without interfering with the existing habitat.

B. Sensor Board

To provide meaningful data to scientists, we designed and manufactured an environmental monitoring sensor board, shown in Figure 2. Named the Mica Weather Board, it provides sensors that monitor changing environmental conditions with the same functionality as a traditional weather station. The Mica Weather Board includes temperature, photoresistor, barometric pressure, humidity, and passive infrared (thermopile) sensors.

Each sensor was chosen from a list of possible candidates with similar characteristics. The barometric pressure module is a digital sensor manufactured by Intersema. The sensor is sensitive up to 0.1 mbar of pressure and measures the absolute pressure range from 300 to 1100 mbar. The module is calibrated during manufacturing and the calibration coefficients are stored in EEPROM on the module. The pressure module includes a calibrated temperature sensor so that the barometric pressure readings may be temperature compensated.

The humidity sensor is manufactured by General Eastern. It is a polymer capacitive sensor factory calibrated to within 1 picofarad ($\pm 3\%$ relative humidity). The sensing element consists of an electrode metalization deposited over the humidity sensor polymer. The sensor is modulated by a 555 CMOS timer to sense the charge in the capacitor which is filtered through by RC circuit. The resulting voltage is amplified by an instrumentation amplifier for greater sensitivity over the range of 0% to 100% relative humidity.

The thermopile is a passive infrared sensor manufactured by Melexis. Heat from black bodies in the sensor’s field of view causes a temperature difference between the thermopile’s cold junction and the thermopile membrane. The temperature difference is converted to an electric potential by the thermo-electric effect in the thermopile junctions. The sensor does not require any supply voltage. The thermopile includes a thermistor in the silicon mass. The thermistor may be used to measure the temperature of the cold junction on the thermopile and accurately calculate the temperature of the black body.

The photoresistor is a variable resistor in a voltage divider circuit. The divided voltage is measured by the ADC. The final temperature sensor is a digital calibrated sensor that communicates over the I²C bus. The characteristics of each sensor can be seen in Table I.

The sensors were chosen with great care to ensure low interchangeability and high accuracy. Each sensor has less than 3% variation when interchanged with others of the same model. The accuracy of each sensor is within 3% of the actual value. Through calibration, the interchangeability and accuracy can be reduced to below 1% depending

Sensor	Accuracy	Interchangeability	Sample Rate (Hz)	Current (in mA)
Photoresistor	N/A	10%	2000	1.235
I ² C Temperature	1 K	0.20 K	2	0.150
Barometric Pressure	1.5 mbar	0.5%	10	0.010
Barometric Pressure Temp	0.8 K	0.24 K	10	0.010
Humidity	2%	3%	500	0.775
Thermopile	3 K	5%	2000	0.170
Thermistor	5 K	10%	2000	0.126

TABLE I

Mica Weather Board: CHARACTERISTICS OF EACH SENSOR INCLUDED ON THE MICA WEATHER BOARD.

on the requirements of the application. Higher accuracy results in a longer time to deploy the nodes due to calibration. Out of the box, the nodes will be accurate for most applications. Due to the interchangeability and accuracy, the sensors can be deployed in the field quicker and little or no calibration is needed prior to deployment.

The unique combination of sensors can be used for a variety of aggregate operations. The thermopile may be used in conjunction with its thermistor and the photoresistor to detect cloud cover [5]. The thermopile may also be used to detect occupancy, measure the temperature of a nearby object (for example, a bird or a nest), and sense changes in temperature in the object over time. If the initial altitude is known, the barometer module may be used as an altimeter. Strategically placed sensor boards with barometric pressure sensors can detect the wind speed and direction by modeling the wind as a fluid flowing over a series of apertures (one such method is described in [6]).

In addition to the sensors on the Mica Weather Board, we included an I²C analog to digital converter. Separating the ADC from the main Mica processing board provides greater flexibility in developing components to reduce power consumption. The ADC uses less power than the Atmel processor on the Mica, may be used in parallel with processing or radio transmission on the Mica, and can be operated in various low-power and sleep modes. Additionally, The sensor board includes an I²C 8 by 8 power switch permitting individual components on the board to be turned on or off. Each switch can be operated independently of each other—further reducing power consumption.

The Mica Weather Board was designed with interoperability in mind. The Mica includes a 51 pin expansion connector. The connector has the ability to stack sensor boards on top of each other. Instead of allowing each board to compete for pins on the connector, we developed an access protocol. The Mica will change the value of a switch on the sensor board using the I²C bus. By mon-



Fig. 2

Mica Hardware Platform: THE MICA SENSOR NODE (TOP LEFT) WITH THE MICA WEATHER BOARD DEVELOPED FOR ENVIRONMENTAL MONITORING APPLICATIONS

itoring the status of the switch, the sensor board knows when access to the Mica's resources has been granted. When a board has access, it may use the power, interrupt, ADC, and EEPROM lines that are directly connected to the microprocessor and components on the Mica processing board.

C. Energy budget

Habitat monitoring applications need to run for nine months. Mica runs on a pair of AA batteries, with a typical capacity of 2.5 ampere-hours (Ah). However we can neither use every drop of energy in the batteries nor are the batteries manufactured with identical capacities from batch to batch or from manufacturer to manufacturer. We make a conservative estimate that the batteries will be able to supply 2200 mAh at 3 volts.

Assuming the system will operate uniformly over the deployment period, each node has 8.148 mAh per day available for use. The application chooses how to allocate this energy budget between sleep modes, sensing, local calculations and communications. We note that since different nodes in the network have different functions, they also may have very different power requirements. For example, nodes near the gateway may need to forward all messages from a patch, whereas a node in a nest may need to merely report its own readings. In any network, there will be some set of power limited nodes; when these nodes exhaust their supplies, the network is disconnected and inoperable. Consequently, we need to budget our power with respect to the energy bottleneck of the network. To form an estimate of what is possible on a Mica mote with a pair of AA batteries, we tabulated the costs of various basic operations in Table II.

Operation	nAh
Transmitting a packet	20.000
Receiving a packet	8.000
Operating sensor for 1 sample (analog)	1.080
Operating sensor for 1 sample (digital)	0.347
Reading a sample from the ADC	0.011
EEPROM Read Data	1.111
EEPROM Program/Erase Data	83.333

TABLE II
POWER REQUIRED BY VARIOUS MICA OPERATIONS.

The baseline life time of the node is determined by the current draw in the sleep state. Minimizing power in sleep mode involves turning off the sensors, the radio, and putting the processor into a deep sleep mode. Additionally, I/O pins on the microcontroller need to be put in a pull-up state whenever possible, as they can contribute as much as 100 μ A of leakage current. Mica architecture uses a DC booster to provide stable voltage from degrading alkaline batteries. With no load, the booster draws between 200 and 300 μ A, depending on the battery voltage. While this functionality is crucial for predictable sensor readings and communications, it is not needed in the

sleep mode. Furthermore, the current draw of the micro-processor is proportional to the supply voltage. We modified Mica motes with a Schottky diode, which allows us to reliably bypass the DC booster while reducing the supply voltage in sleep modes. The modification allows us to achieve between 30 and 50 μ A current draw (battery dependent), which reduces the energy available for tasks to 6.9 mAh per day.

D. Electro-mechanical Packaging

By their nature, habitat monitoring sensors are exposed to the environment. Packaging for environmental sensors must protect the device, while minimally obstructing the sensing function. Mica motes by their design are fairly robust mechanically, with the battery case firmly integrated with the main sensor board, and the mounting holes for securing the sensor boards. To provide weather-proofing, we coat the entire sensor package with paralene sealant, which protects all exposed electrical contacts from exposure to water. The sensors remain exposed to protect their sensitivity. Each coated node is then enclosed in a transparent acrylic enclosure. The enclosure must be ventilated again as not to distort the sensor readings; its primary function is to provide additional protection against mechanical failures and to raise the sensor off the ground. Acrylic packaging was chosen because it is infrared and radio frequency transparent, which won't obstruct sensor readings or wireless communication.

E. Patch Gateways

We chose CerfCube, a small, StrongArm-based embedded system, to act as the sensor patch gateway. Each gateway is equipped with a CompactFlash-based 802.11b adapter. Porting functionality to CerfCubes is fairly easy; they run an embedded version of Linux operating system. Permanent storage is plentiful—the gateway can use the IBM MicroDrive which provides up to 1 GB of storage. Supplying adequate power for this device is a challenge, without power management features this device consumes about 2.5W (two orders of magnitude more than the sensor nodes). Currently we're considering a solar panel providing between 60 and 120 Watts in full sunlight connected to a rechargeable battery with capacity between 50 and 100 Watt-hours (*e.g.* sealed lead-acid). Researchers from Intel Research and JPL have demonstrated delay-tolerant networking using CerfCubes and motes[3] which will fit very well with the overall system architecture.

F. Base-station installation

In order to provide remote access to the habitat monitoring networks, the collection of sensor network patches is

connected to the Internet through a wide-area link. James Reserve is already equipped with a T1 line. On Great Duck Island, we connect to the Internet through a two-way satellite connection provided by DirecWay. The satellite system is connected to a laptop which coordinates the sensor patches and provides a relational database service. We had to solve a number of challenges to turn a consumer-grade, web-oriented service into a highly reliable general-purpose network connection. The base station needs to function as a turnkey system, since it needs to run unattended and during that time we do expect unscheduled system reboots. At this point we have resolved many of the engineering issues surrounding this problem – shortly after the system boots we can find it on the Internet and access it remotely.

G. Database Management System

The base station currently uses Postgres SQL database. The database stores time-stamped readings from the sensors, health status of individual sensors (*e.g.* battery status) and the network as a whole (*e.g.* connectivity and routing information) as well as metadata (*e.g.* sensor locations).

H. User Interfaces

We expect that many user interfaces will be implemented on top of the sensor network database. GIS systems provide a widely used standard for analyzing geographical data. Most statistics and data analysis packages, such as Matlab, implement powerful interfaces to relational databases. Finally, we expect a number of web based interfaces to provide the ubiquitous interfaces to the habitat data.

At this point, the gizmo design for local users is not well developed. We expect that the design will be based around an iPaq PDA running Linux. The device will interface with the mote network through a CompactFlash-based interface to the mote [7]. A second CompactFlash slot can be used to connect to the wireless LAN.

V. SENSOR NETWORK SERVICES

All of the components in the system must operate in accordance with the system's power budget. As we pointed out in section IV, each node has a budget of 6.9 mAh per day. Since Mica processor alone draws approximately 5 mA, we can afford to run the processor for at most 1.4 hours per day, 5.8% duty cycle if **no** other operations are performed by the mote. In a running system, the energy budget must be divided amongst several system services: sensor sampling, data collection, routing, health monitoring and network retasking.

A. Data sampling and collection

In habitat monitoring the ultimate goal is data collection; sampling rates and precision of measurements are often dictated by external specifications. For every sensor we can bound the cost of taking a single sample. By analyzing the requirements we can place a bound on the energy spent on data acquisition. We trade the cost of data processing and compression against the cost of data transmission. We can estimate the energy required by data collection by analyzing data collected from indoor monitoring networks. Let us consider an experiment where a mote collects a light sample every minute. The sample is represented as a 16-bit integer, but it contains a 10-bit ADC reading. Assuming that each packet can carry 25 bytes of payload, unprocessed data requires between 72 (if 10-bit samples are used) and 116 packets (if 16-bit numbers are used). While this service does not put a burden on the leaf nodes, the routing nodes near the root may need to retransmit the messages from every leaf in the network, roughly two orders of magnitude more. Anecdotal evidence presented Table III suggests that this volume of data can be easily reduced by a factor of 2-4 by applying a delta compression and a standard compression algorithm (*e.g.* Huffman coding or Lempel-Ziv). The compression performs even better when applied to a longer run of data. Far better results can be obtained with signal-specific lossy compression techniques (much like the GSM voice compression schemes). Other methods include distributed compression involving correlating network data amongst similar nodes and using Coset codes [8]. Often the signal model is unknown a priori, but can be obtained through the analysis of the initial data. We can then use the network retasking service to program the sensors to communicate the data of interest.

Once we have allocated the energy for sampling the sensor and communicating the results, the remaining energy is devoted to maintaining the network – MAC protocols, maintaining routing tables, forwarding network messages, and health monitoring. These tasks can either be tightly scheduled or run on demand. On one extreme, the system is scheduled at every level, from TDMA access to the channel, through scheduled adaptation of routes and channel quality. Overhead costs are upfront and fixed. A TDMA system is expected to perform well if the network is relatively static. On the other extreme, we use a low-power hailing channel to create on-demand synchronization between a sender and a receiver. The service overhead is proportional to the use of the service. This approach can be more robust to unexpected changes in the network, at the expense of extra cost. Finally, a hybrid approach is possible, where each service runs in an on-

Compression algorithm	Huffman (pack)	Lempel-Ziv (gzip)	Burrow-Wheeler (bzip2)	Uncompressed
8-bit sample	1128	611	681	1365
10-bit sample	1827	1404	1480	1707
16-bit sample	2074	1263	1193	2730
8-bit difference	347	324	298	1365
10-bit difference	936	911	848	1707
16-bit difference	839	755	769	2730

TABLE III

COMPRESSION CHARACTERISTICS OF TYPICAL INDOOR LIGHT SIGNAL. WE ESTIMATE THE AMOUNT OF INFORMATION CONTAINED WITHIN THE SIGNAL BY COMPRESSING VARIOUS SIGNAL REPRESENTATIONS WITH THE STANDARD UNIX COMPRESSION UTILITIES.

demand fashion, but the time period for when the demand can occur is scheduled on a coarse basis.

B. Communications

The communications service consists of the communications resources including hardware and a set of routing and media access algorithms. The routing algorithms must be tailored for efficient network communication while maintaining connectivity when required to source or relay packets.

A simple routing solution for low duty cycle sensor networks is simply broadcasting data to a gateway during scheduled communication periods. This method is the most efficient—data is only communicated in one direction and there is no dependency on surrounding nodes for relaying packets in a multihop manner.

Many of the hard to reach research locations are beyond the range of a single wireless broadcast from mote to gateway. Accordingly, a multi-hop scheduled protocol must be used to collect, aggregate, and communicate data.

Methods like GAF [9] and SPAN [10] have been used to extend the longevity of the network by selecting representatives to participate in the network; thereby these algorithms reduce the average per node power consumption. Although these methods provide factors of 2 to 3 times longer network operation, our application requires a factor of 100 times longer network operation. GAF and SPAN don't account for infrequent sampling but rather continuous network connectivity and operation. Instead, we propose augmenting scheduled multihop routing or low power MAC protocols with GAF and/or SPAN to provide additional power savings. GAF and SPAN are independent of sampling frequency, whereas our application requires increased power savings that may be achieved by adjusting the communication frequency.

The research challenge of the routing problem is finding a power efficient method for scheduling the nodes such that long multihop paths may be used to relay the

data. We propose the following approaches for scheduled communication:

- After determining an initial routing tree, set each mote's level from the gateway. Schedule nodes for communication on adjacent levels starting at the leaves. As each level transmits to the next, it returns to a sleep state. The following level is awoken, and packets are relayed for the scheduled time period. The process continues until all levels have completed transmission in their period. The entire network returns to a sleep mode. This process repeats itself at a specified point in the future.
- Instead of a horizontal approach, awaken nodes along paths or subtrees in a vertical approach. Each subtree in turn completes their communication up the tree. This method is more resilient to network contention; however the number of subtrees in the network will likely exceed the number of levels in the network and subtrees may be disjoint allowing them to communicate in parallel.

Alternatively, we have experimented with using low power MAC protocols. By determining our duty cycle, we can calculate the frequency with which the radio samples for a start symbol. By extending the start symbol when transmitting packets, we can match the length of the start symbol to the sampling frequency. Other low power MAC protocols, such as S-MAC [11] and Aloha [12] employ similar techniques that turn off the radio during idle periods to reduce power consumption. The difference is that instead of having a large power and network overhead of setting up a schedule initially, the overhead is distributed along the lifetime of the node. Both approaches are equivalent in power consumption, the decision for which to use depends on the end-user interactivity required by the application. A potential tradeoff of using a low power MAC is that transmitted packets potentially wake up every node within the cell. Although early rejection can be applied, scheduling prevents unneeded nodes from wasting power

processing a packet's headers.

C. Network Retasking

As the researchers refine the experiment, it may be necessary to adjust the functionality of individual nodes. This refinement can take several different forms. Scalar parameters, like duty cycle or sampling rates, may be adjusted through the application manager. Even such simple adjustment allows the researchers to focus their efforts in more interesting areas. Most of the time such updates can be encapsulated in network maintenance packets. More complex functionality adjustment may be implemented through virtual machines like Maté [13]. Virtual machine-based retasking seems ideal when the much of the underlying functionality is implemented through underlying native functions, as is the case in making routing decisions, or processing data through a predefined set of filters. Virtual machine programs can be fairly small (many fit in a single packet). Finally, the entire code image running on a mote may be replaced with a new one. One would use this method when a drastic retasking of the application is necessary; for example if it were necessary to install a new signal-specific compression algorithm to cope with the volume of data. The reprogramming process is quite costly – it involves reliably transmitting the binary image of the code (transmission on the order of 10kbps of data) to all nodes that need to be reprogrammed, and invoking a reprogramming application which runs the node for about 2 minutes while drawing about 10 mA. To relate this to the energy budget: we can afford to reprogram the nodes every day during the 9 month life cycle if reprogramming is the node's only task. While significantly more expensive in absolute terms than virtual machine reprogramming, it can pay off over the period of a few days since it can execute code more efficiently.

D. Health and Status Monitoring

A major component of use to the application is one that monitors the mote's health and the health of neighboring motes. Health and monitoring is essential for a variety of purposes; the most obvious is retasking. The duty cycle of a mote may be dynamically adjusted to alter the lifetime of a particular mote.

Health and monitoring messages sent to the gateway can be used to infer the validity of the mote's sensor readings. Although the health messages are not critical for correct application execution, their use can be seen as preventive maintenance. For this reason, we implement a health and monitoring component that does not rely on reliable transport like other data (such as the mote's log or

data summary statistics), but ensures low latency. Health messages are sent rather infrequently (about once per hour dependent on the duty cycle) with no guarantee on their delivery.

VI. CURRENT PROGRESS

We have deployed two small scale sensor networks in James Reserve and Great Duck Island. These systems have nearly all core architecture components described in section III including Mica nodes, sensor boards, weather resistant packaging, base station, relational database, and wide area connectivity. We plan to add an intermediate tier of WLAN connectivity to multiple sensor patches this summer.

The initial deployment of motes at GDI and JMR did not include any calibration among the sensors. In order to provide greater accuracy and consistency amongst sensor readings, we feel that developing an calibration or auto-calibration procedure would be a useful tool for establishing a reliable sensor network in field applications.

Our current focus is on energy efficient strategies for multihop routing. In the next few weeks we will evaluate the globally scheduled communication and the demand-driven low-power MAC. We are confident that we will be able to quantify the design tradeoffs and extend the currently deployed prototypes with the services described in section V.

Our intention is to develop and package a habitat monitoring kit. This kit will be completed in six months and made available to scientists and researchers. Our goal is to tackle the technical problems and meet the application requirements set in section II through the proposed architecture in section III and services in section V. The creation of this kit will allow scientists to reliably collect data from locations previously inaccessible. The data is made available to scientists through the data store and interactive devices (such as the gizmo or other user interface), effectively abstracting the underlying technical details of the system.

VII. CONCLUSION

Habitat and environmental monitoring represent an important class of sensor network applications. We are collaborating with biologists at the College of the Atlantic and the James Reserve to define the core application requirements. Because the end users are ultimately interested *only* in the sensor data, the sensor network system must primarily deliver the data of interest in a confidence-inspiring manner. The low-level energy constraints of the sensor nodes combined with the data delivery requirements leave a clearly defined energy budget for all other

services. Tight energy bounds and the need for predictable operation guide the development of application architecture and services.

REFERENCES

- [1] Deborah Estrin, Ramesh Govindan, John S. Heidemann, and Satish Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Mobile Computing and Networking*, 1999, pp. 263–270.
- [2] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, UT, May 2001.
- [3] Kevin Fall, "Delay-tolerant networking for extreme environments," <http://www.cs.berkeley.edu/~kfall/extreme-talk.pdf>, Nov. 2001, Presentation at UCSD.
- [4] Jason Hill and David Culler, "A wireless embedded sensor architecture for system-level optimization," in *Submission to USENIX ASPLOS '02*, 2002.
- [5] R. W. Clay, N. R. Wild, D. J. Bird, B. R. Dawson, M. Johnston, R. Patrick, and A. Sewell, "A cloud monitoring system for remote sites," *Publications of the Astronomical Society of Australia*, vol. 15, no. 3, pp. 332–335, Aug. 1998.
- [6] John D.W. Barrick, John A. Ritter, Catherine E. Watson, Mark W. Wynkoop, John K. Quinn, and Daniel R. Norfolk, "Calibration of NASA turbulent air motion measurement system," NASA Technical Paper 3610, Langley Research Center, Dec. 1996.
- [7] Thanos Stathopoulos, "MoteNIC: Overview," http://lecs.cs.ucla.edu/Noteworthy/quadcharts/thanos_lecs.ppt, Feb. 2002.
- [8] Julius Kusuma, Lance Doherty, and Kannan Ramchandran, "Distributed compression for wireless sensor networks," in *Proceedings of ICIP 2001*, Thessalonika, Greece, Oct. 2001.
- [9] Ya Xu, John Heidemann, and Deborah Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001, USC/Information Sciences Institute, pp. 70–84, ACM.
- [10] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001, pp. 85–96.
- [11] Wei Ye, John Heidemann, and Deborah Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, USA, June 2002.
- [12] Amre El-Hoiydi, "Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks," in *Proceedings of IEEE International Conference on Communications*, New York, NY, USA, Apr. 2002.
- [13] Philip Levis and David Culler, "Maté: A tiny virtual machine for sensor networks," in *International Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, CA, USA, Oct. 2002, To appear.